웹 기반 소프트웨어의 시험 및 검증 기술

동국대학교 최은만*

1. 서 론

웹과 인터넷이 보급되면서 일반인이 컴퓨터에 더 가깝게 되었고 인트라넷의 구축으로 웹 환경에 사용할 목적으로 개발되는 소프트웨어가 늘어나고 있다. 특히 E-비즈니스의 확장과 함께 웹 기반 소프트웨어의 개발과 시험은 비즈니스 성패에 중요한 요소이다. 웹사이트의 내용과 사이트를 구성하는 웹 기반 소프트웨어는 전 세계에 흩어져 있는 수많은 사용자들을 대상으로 한다는 면에서 품질과 시험 방법 또한 중요하다[1].

웹기반의 소프트웨어는 일반 소프트웨어와는 다르게 다양한 요소로 구성되어 있다. 분산 시스템의 프론트엔드인 클라이언트에는 웹 브라우저와 플러그인, 내장된 객체 등이 있으며 백엔드에는 응용 서버의 컴포넌트를 비롯하여 데이터베이스 컴포넌트 및 다양한 3th-party 모듈이 포함될 수 있다. 따라서 웹 기반 소프트웨어를 이루는 다양한 요소들을 시험하고 검증하기 위하여 다양한 방법이 동원되고 있다.

또한 웹 기반 소프트웨어를 시험하는 관점이나 검증할 사항은 웹 기반이 아닌 일반 소프트웨어와 크게 다르다. 브라우저와의 호환성, 네비게이션 오류, 웹 UI, 시스템 통합, 데이터베이스, 보안, 반응시간, 스트레스 시험 등이 시험에서 중점적으로 다루어져야 한다.

이 논문에서는 웹 기반 소프트웨어의 구성요소와 품질 특성이 어떻게 차이가 나며 고유한 특성을 시험하고 평가하는 기술과 도구가 어떤 것이 있는지 소개한다.

2. 웹 기반 소프트웨어의 구성요소

웹 기반 소프트웨어는 일반 소프트웨어와 여러면에서 다르다. 인쇄 출판과 소프트웨어 개발, 마케팅과 컴퓨팅, 내부 통신과 외부 관계, 예술과 기술이 혼재되어 있다[2]. 단순한 응용 프로그램보다는 다양한 컨텐트가 존재한다는 특징이 있다. 자원의 공유로 시작한 웹이 최근 E-비즈니스로 발전하면서 구성 요소도 다양하게 되었다. 웹 시스템은 간단한 텍스트로부터 그래픽, 오디오, 비디오 컨텐트 등의 공유하기 위한 자원뿐만 아니라 분산 시스템을 이루는 다양한 소프트웨어 컴포넌트들이 존재한다. 웹기반 클라이언트-서버 시스템은 일반적으로 그림 1과 같이 세 가지 계층의 컴포넌트로 나눌 수 있다.

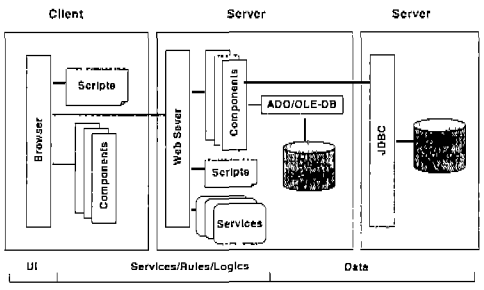

그림 1 3-계층 웹 기반 소프트웨어의 구조[3]

클라이언트에는 웹 브라우저, 스크립트, 플러그 인 컴포넌트 등 사용자 서비스 컴포넌트들이 있고 서버에는 비즈니스 로직이 있는 응용서버 컴포넌트가 있으며 데이터 저장을 위하여 데이터 서비스 컴포넌트가 있다. 클라이언트에 있는 웹 브라우저는 웹 서버에게 웹 페이지들을 요청하며 웹 페이지 안에 있는 HTML, ASP, DHTML 등의 코드는 웹 브라우저에게 웹페이지를 디스플레이 하는 방법을 지시한다.

* 종신회원

Java, ActiveX, 스크립트 언어인 JavaScript, VB Script로 작성된 내용도 포함되며 HTML, GIF, JPEG 이외의 사운드 및 비디오 데이터를 처리하기 위한 플러그 인 컴포넌트도 있다. 반면 서버 측에는 웹페이지를 저장하고 이를 클라이언트에 제공하는 웹 서버와 웹 어플리케이션을 위한 자료 저장소를 제공하는 데이터베이스 서버, 웹 서버의 기능과 데이터베이스 서버를 잘 연결시키는 어플리케이션 서버가 있다. 웹을 구성하는 다양한 소프트웨어 컴포넌트들을 정리하면 다음 표 1과 같다.

표 1 웹 기반 소프트웨어 컴포넌트의 분류

클라이언트 컴포넌트	서버 컴포넌트	3-party 컴포넌트
- 웹 브라우저 - 스크립트 - Java VM - 플러그 인 - 내장된 객체들	- 웹 서버 - 스크립트 - Java VM - 데이터베이스 서버 - 데이터 액세스 서비스 - 트랜잭션 서비스	- Java 컴포넌트 - ActiveX - Standard EXE - Standard DLL - CGI

이와 같은 다양한 컴포넌트로 구성된 그림 2와 같은 사용자 인터페이스를 가진 웹 기반 소프트웨어를 상상해 보자. 구성요소는 서버에서 수행되는 데이터베이스 응용뿐만 아니라 클라이언트에서 수행된 HTML 페이지 안에서 호출된 Java 기반의 차트를 생성하는 애플릿이 있다고 가정한다. 이렇게 분산된 다양한 컴포넌트가 하나의 웹 기반 소프트웨어를 이루고 있으므로 각 컴포넌트뿐만 아니라 컴포넌트 사이의 연결도 시험의 대상이 된다.

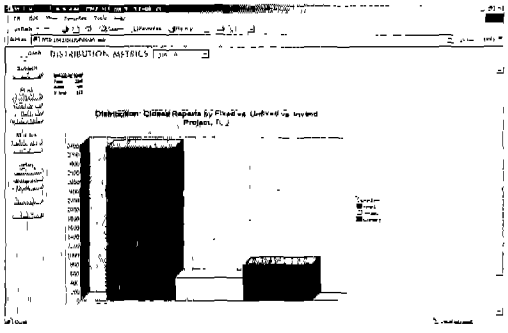

그림 2 Java 애플릿이 포함된 웹 기반 소프트웨어

그림 2의 웹 기반 소프트웨어를 구성하는 여러 컴포넌트의 실행 흐름은 그림 3과 같다. 이러한 실행 흐름은 웹 기반 소프트웨어의 시험 사례를 설계하는데 매우 중요하다.

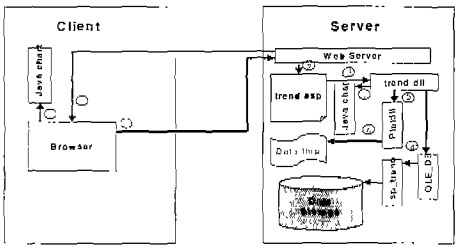

그림 3 웹 컴포넌트의 실행 흐름

① 사용자가 trend 차트를 생성하도록 요청
② 웹 서버가 trend.asp 요청
③ trend.dll이 호출됨
④ trend.dll이 데이터베이스 서버와 연결되어 요청한 자료를 찾을 sp_trend라는 프로시저를 호출
⑤ trend.dll이 데이터베이스에 요청한 자료를 받아 plot.dll에 넘겨 계산하고 차트를 작성
⑥ 포매팅된 최종 자료가 data.tmp에 저장
⑦ 차트 작성 Java 컴포넌트가 호출되어 data.tmp를 이용하여 차트 작성
⑧ Java 애플릿이 클라이언트에 전송됨
⑨ Java 애플릿이 사용자 브라우저에 로드되어 계산된 자료가 차트로 그려짐

3. 품질 및 시험기술의 특성

웹 기반 소프트웨어는 품질과 신뢰성 측면에서도 일반 소프트웨어와 다르다. 일반 소프트웨어와 비교하여 다음과 같은 품질 특성을 가지고 있다.

- 변경이 많음 - 웹사이트의 내용은 최신의 것으로 유지되어야 하는 것이 많다. 따라서 유지보수가 빈번히 일어난다. 웹 페이지의 99.1%가 1개월 안에 변경된다[4].
- 구조적 품질 - 웹사이트를 구성하는 여러 링크들이 잘 연결되어 모두 작동하는가를 나타내는 품질. 즉 네비게이션 테스트가 필요하다.
- 컨텐트의 품질 - 웹 페이지에는 다양한 형태의 미디어가 존재한다. 매우 빈번히 변경이 이루어지거

나 동적으로 생성되는 페이지의 경우 그림, 음성, 화상 등의 컨텐트가 잘 연결되었는지를 뜻한다.

● 반응 시간 - 웹 시스템은 네트워크로 연결된 클라이언트 서버 시스템이므로 브라우저에서 요청한 작업의 결과가 작업을 방해하지 않는 짧은 시간 내에 완성되어야 한다.

● 정확 및 일관성 - 데이터베이스와 연동되어 동적인 HTML을 생성하는 경우 일관성 있고 정확하여야 한다.

결국 웹 기반 소프트웨어를 시험하는 방법은 앞서 설명한 웹 기반 소프트웨어의 구성요소들을 시험할 수 있어야 하며 통합하였을 때 위와 같은 품질 특성들을 평가할 수 있어야 한다.

4. 웹 기반 소프트웨어의 시험 기술

소프트웨어를 시험할 때 동원하는 기술은 각기 중점적으로 체크하려는 분명한 목표가 있다. 예를 들면 통합 시험 방법은 모듈의 인터페이스를 시험하려는 목적이 있으며 스트레스 시험은 소프트웨어의 최대 자료처리량을 점검하기 위한 것이다.

웹 테스트에 관한 여러 가지 연구와 경험에 의하면 다음과 같은 시험 기술이 도입되고 있다.

4.1 사용자인터페이스 테스트

사용자 인터페이스 테스트에서는 다음 세 가지 사항을 점검한다.

● 설계 기본 방향 - 웹 페이지를 설계할 때 중요한 점은 메뉴, 입력창, 오류창, 결과 등이 사용자에게 일관적인 모양과 위치를 갖도록 하는 것이다. 사용자 인터페이스 설계에 메타포(예를 들면 웹 페이지의 메뉴가 도서관의 책 분류 색인을 상상하도록 하는 개념)를 도입하기도 하는데[5] 적당한 메타포가 선택되었는지, 사용자가 예상하지 못한 사용자 인터페이스의 선택 또는 반응이 없는지 시험하고 점검한다.

● 사용자 입력 - HTML 페이지에서 사용자 입력과 관련된 요소, 즉 폼, 라디오 버튼, 체크 박스, 텍스트 필드, 스크롤, 메뉴 등을 체크한다. 특히 동적 UI 콘트롤을 위하여 작성된 여러 가지 스크립트와 자바 애플릿, CGI 컨트롤들이 잘 연결되어 있고 브라우저와 호환성이 있는지 체크하여야 한다.

● 자료의 표현 - 웹 브라우저에 출력된 자료의 오류는 (1) 데이터 자체의 오류와 (2) 데이터베이스 질의어 오류, (3) 브라우저 표현 오류로 나눌 수 있다. 오류의 종류에 따라 해당되는 부분, 즉 데이터베이스 자체, 질의어, 서버 측 스크립트 등을 점검한다.

4.2 네비게이션 테스트

네비게이션 오류를 찾아내기 위하여 웹 설계 모델을 검토하고 링크의 오류를 찾아낸다. 끊어진 링크, 옮겨진 페이지, 고립된 페이지 등을 찾아내기 위하여 웹사이트에 연결된 모든 링크를 네비게이션한다. SilkTest[6], Doctor HTML[7]과 같은 도구들은 기계적인 작업을 자동화한 것이다.

4.3 브라우저 렌더링 테스트

브라우저는 웹 응용이 실행되는 환경으로 웹 페이지에 사용된 브라우저의 종류와 버전마다 렌더링과 지원하는 기능이 다르다. 따라서 웹 페이지 안에 포함된 스크립트, 애플릿, ActiveX 컨트롤, 스타일, HTML, 플러그인 등이 브라우저와 호환성이 있는지 점검하여야한다.

4.4 기능 테스트

소프트웨어 내부에 대한 자세한 정보 없이 테스트 입력을 소프트웨어 기능별로 잘 준비하여 시험하는 방법은 기능 테스트라 한다. 일반적인 소프트웨어에 적용할 수 있는 기능 테스트 방법들이 웹 기반 소프트웨어에도 모두 적용될 수 있다. 기능 테스트 방법은 시험 대상 범위를 어떻게 정의하느냐에 따라 표 2와 같이 테스트 데이터 작성이 달라진다.

4.5 데이터베이스 테스트

데이터베이스에 보관된 웹기반 소프트웨어의 데이터인 컨텐트에 이상이 없는지, 스키마가 제대로 되었는지 시험한다. 대부분의 웹 기반 소프트웨어들이 SQL을 사용하여 데이터베이스와 연동하므로 SQL을 이용하여 데이터가 잘 보관되었는지 확인한다.

데이터 자체의 확인뿐만 아니라 웹 기반 소프트웨어를 이루는 컴포넌트 사이의 인터액션이 잘 이루어지는지 시험하여야 한다. 그림 4에 나타낸 것이 데이터베이스 테스트를 위한 인터액션 포인트들이다.

데이터베이스와 관련된 오류는 대부분 자료의 일관성에 대한 오류와 출력의 오류이다.

표 2 기능 테스트의 종류

종 류	방 법	시험 내용
단순 기능 테스트	입력 기능과 네비게이션이 제대로 작동하는지 점검. 웹 페이지나 윈도우의 사용자 인터페이스를 이루는 텍스트 박스, 풀 다운 리스트, 라디오 버튼 등 여러 가지 컨트롤의 작동여부 검토. 또한 입력 된 후 액션버튼의 선택에 따라 데이터가 제대로 전달되어 작업이 이루어지는지 확인하는 수준.	입력 필드의 경계값 처리가 되는가? 처리결과가 원하는 우선순위대로 출력되는가?
태스크 중심 기능 테스트	설계 명세서를 자세히 분석하여 기능 태스크 별로 분해하고 성능의 경쟁력이 있으며 시장 요구에 적합한 것인지 점검.	구현된 기능이 과연 명세에 있는 것이며 유용한 것인가?
오류주입 테스트	1. 오류조건 주입 2. 오류인식 로직 점검 3. 오류처리 로직 점검 4. 오류 메시지의 출력과정 점검	의도적으로 오류를 주입하여 개발할 때 미처 예상하지 못한 오류조건을 파악.
경계값 및 동치클래스 테스트	입력값의 범위에 대한 동치 클래스를 작성하여 각 클래스의 대푯값과 경계값에 대하여 테스트 사례를 작성	입력 자료의 자릿수, 범위를 점검
탐구 테스트	프로그램에 대한 자극이라 할 수 있는 입력을 주고 그 반응을 보아 다음 테스트 사례를 결정	계획된 테스트 사례로 커버할 수 없는 부분을 점검

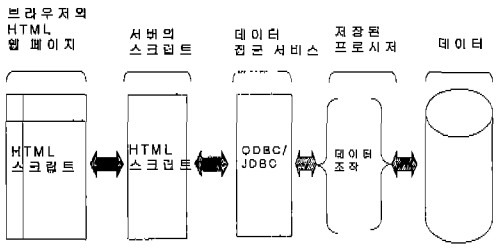

그림 4 데이터베이스 테스트 시험을 위한 인터액션 포인트

자료가 보관되어 있지 않거나 보관되어 있지만 여러 가지 이유로 정확한 자료가 웹브라우저의 사용자에게 보여지지 않는 경우가 있다. 자료의 타입을 잘못 취급하거나 필드의 크기를 초과하거나 테이블 선언이 잘 못 되었거나 저장 과정에 오류가 있을 때 이러한 현상을 보인다.

데이터베이스 오류를 찾아내기 위하여 사용하는 방법은 SQL 문장이나 저장 프로시저를 하나씩 수행해보는 화이트박스 시험과 웹브라우저 인터페이스를 통하여 직접 입력하여 백엔드 데이터베이스에 전달되고 저장되는 데이터를 확인하는 블랙박스 시험 방법이 있다.

4.6 Help 테스트

소프트웨어의 사용용이성을 시험하기 위한 중요한 요소가 소프트웨어의 Help 기능이다. 웹기반 Help 시스템의 구성과 설계, 용어, 폰트, 포맷에 일관성이 있는지 체크하고 과연 Help 기능이 쓸모가 있는지 점검하며 기능 오류(점프, 팝업, 버튼, 네비게이션, 프레임 등에 대한 오류)가 없는지 살핀다.

4.7 보안 테스트

쇼핑몰이나 경매 등 전자상거래가 가능한 사이트에서는 무엇보다 보안이 중요하다. 보안 시스템은 소프트웨어뿐만 아니라 구성인력, 통신 장비 등 여러 가지 요소로 이루어지므로 소프트웨어의 보안 테스트만으로 100퍼센트 안전한 시스템이 될 수 없지만 웹 기반 소프트웨어가 보안 취약점이 없는지 살펴보는 것이 첫째로 필요한 작업이다. 특히 HTML에 포함된 Java 코드들이나 ActiveX 콘트롤들은 클라이언트 머신에서 수행되므로 로컬 드라이브에 있는 파일을 악의로 삭제하거나 중요한 정보를 가져갈 수 있다. 또한 클라이언트 브라우저 메모리나 머신에 남아있는 쿠키정보들을 악의가 있는 웹사이트 서버에서 읽어낼 수가 있다. 또한 바이러스, 웜, 서비스가 불가

표 3 보안 테스트 방법과 대상

액티비티	보호			보호 목표					보안 방법				
	사용자	제작자	타입	클라이언트/서버 응용	데이터베이스	서버	클라이언트	네트워크	방화벽	암호화	인증	접근제어	데이터 보호
안티 바이러스	✓	✓	데이터	✓		✓	✓		✓			✓	
데이터베이스 접근 보호		✓	데이터		✓						✓	✓	
해킹에 의한 데이터 손실		✓	데이터					✓		✓	✓		
개인 정보 누출 및 삭제		✓	데이터	✓	✓	✓		✓				✓	✓
사설 네트워크의 접근		✓	데이터	✓	✓	✓		✓		✓	✓		
스팸 메일	✓		프라이버시	✓			✓						

능하게 하는 공격 등이 시스템을 위협한다.

웹기반 시스템에서 사용되는 보안 기술은 암호화, 인증, 방화벽 등이 있다. 보안 테스트에서는 클라이언트로부터 DB 서버에 이르는 경로는 추적하여 각 구간마다 보안이 제대로 이루어지고 있는지 체크한다. 다른 방법으로는 실제 시스템에 의도적으로 침입하여 네트워크 시스템이 효과적으로 방어하고 있는지 평가하는 침입 테스트가 있다.

보안테스트를 하려면 먼저 테스트하려는 대상과 목표를 정하고 목표에 맞는 테스트 액티비티를 선택하여야 한다. 표 3은 보안 테스트 액티비티의 종류와 각 액티비티에 의하여 점검될 수 있는 보안 대상, 보안 방법을 표시하였다.

4.8 성능, 스트레스 테스트

웹 기반 소프트웨어가 상업적인 거래로 이어질 때 시스템의 성능은 매출과 직결되는 매우 중요한 문제이다. 예를 들어 하루에 30만 트랜젝션을 처리하고 있다면 1초에 약 3.47개의 트랜젝션을 처리하고 있는 셈이다. 현재의 시스템 반응 속도가 트랜젝션 당 4초를 넘어 10초 미만이 된다면 약 30퍼센트의 사용자가 거래를 하지 못하게 된다. 따라서 성능에 대한 정확한 예측과 병목 현상이 발생되는 곳의 파악이 필요하다. 온라인 트랜젝션을 결정짓는 세 가지 요소, 즉 클라이언트, 네트워크, 서버에 대한 병목 현상과 관련된 자원은 표 4와 같다.

표 4 병목현상을 일으키는 자원과 작업

	브라우저	네트워크	서버
자원 병목 현상	CPU 타임	네트워크 장비에 의한 Latency	CPU 타임 I/O 타임
병목 작업	데이터 전송 데이터 포매팅 데이터 디스플레이 스크립트 및 액티브 컨텐트의 실행	패킷 라우팅	스크립트, 라이브러리함수, 프로시저의 수행

5. 상태기반 시험

트랜젝션 기반 시스템이나 상태 기반 시스템들은 주로 유한 상태로 시스템을 표현하고 이를 테스트에 응용한다. 웹기반 시스템도 상태 기반 시스템으로 간주하고 유한 상태 기계를 이용한 테스트 방법[9]을 적용할 수 있다. 이 방법은 시스템의 행위를 유한상태의 집합으로 나타내고 여기서 접근 가능한 최소의 유한상태기계(minimal finite state machine)를 찾아내어 각 상태를 구동시킬 수 있는 트랜지션들을 찾아내는 것이다.

웹 기반 시스템을 상태 기반으로 테스트하려면 모든 상태들을 커버하는 부분 경로(partial-W)의 집합을 구하고 부분 경로를 구동하는 입력값들을 구며한 된다. 우선 웹 홈페이지에 나타난 시스템의 상태를 FSM으로 나타내어야 하는데 그 방법은 다음과 같다.

표 5 FSM과 웹 페이지 요소 매핑

상태기계의 요소	웹 페이지 요소
상태	웹 응용이 가질 수 있는 모든 상태 (입력 상태, 처리 중, 등등)
트랜지션	웹 페이지에 가해진 자극으로 인하여 이루어지는 웹 페이지의 상태 변환
입력	상태 변환을 일으키는 사용자 자극 (필드 입력, 메뉴 선택)
출력	새로운 상태 변화에 의하여 출력되는 결과

예를 들면 그림 5와 같은 회원가입 기능을 시작으로 멤버 기능, 뉴스, 날씨, 시장 정보, 라이브러리 등이 구현되어 FSM은 웹 응용이 화면에 표현된 것으로 구할 수 있다. 즉 웹 페이지의 상태를 변화시키는 입력과 그 결과인 출력을 찾아내어 그래프로 구성하면 그림 6과 같이 되며 위에서 제시한 최상위 홈 페이지는 Home, My E-mail, News, Weather, Market, Library, Help, Contact Us 등 여러 다른 페이지와 링크되어 있음을 알 수 있다.

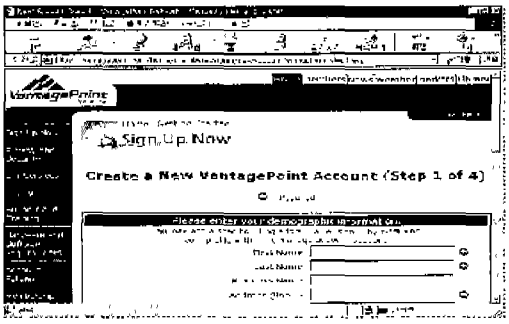

그림 5 등록 홈페이지의 일부

표 6 FSM을 구성하기 위한 시스템 분할

상 태	웹 페이지	근 거
Home	홈페이지	가입신청 전환점
Sign Up Now	Sign Up Now 전반	가입을 위한 신청
Enter Demographic Information	Sign Up Now 중반	인적 정보
Enter Payment Information	Sign Up Now 후반	회비 납부 관련 사항
Enter User name/Password Information	Sign Up Now	시스템 사용 계정에 관한 사항

웹 응용은 그 규모가 커서 단번의 FSM으로 그리기에는 벅차다. 따라서 웹 페이지의 정보를 기초로 하여 적당히 서브 시스템으로 분할할 필요가 있다. 여기서 예로든 VantagePoint 홈페이지의 경우 표 5와 같이 다섯 가지 서브 시스템으로 그루핑할 수 있다

웹 응용을 유한상태기계로 구축할 때 중요한 것은 각 상태로부터 트랜지션할 수 있는 상태를 결정하는 것이다. 웹의 상태 변화를 일으키는 두 가지는 링크와 사용자의 입력이나 버튼을 누르는 행위이다. 표 6에서 분할한 다섯 가지 상태는 그 입력 순서에 따라 다른 상태변화를 가져올 수 있다. 예를 들면 인적 사항을 입력하고 다음으로 회비 납부 정보, 사용자 이름 및 패스워드 순서로 입력할 수도 있고 회비 납부 정보를 먼저 입력하고 다음으로 사용자 패스워드, 마지막으로 인적사항을 입력할 수도 있다. 그림 6은 모든 가능 오퍼레이션의 조합을 FSM으로 나타낸 것이다.

FSM으로 표현한 모든 상태는 웹페이지에서 이루어진 사용자 액션을 기초로 도달할 수 있는 상태를

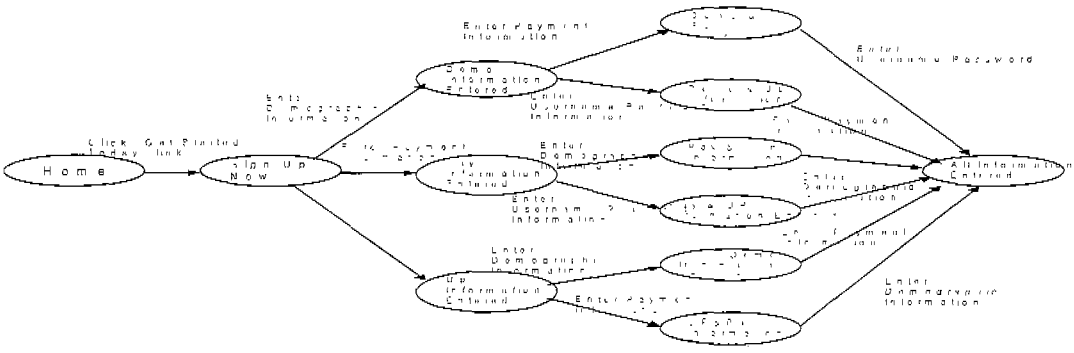

그림 6 사용자 등록을 위한 상태변환

의미한다. 따라서 웹 테스트를 위하여 필요한 것은 바로 사용자 액션, 즉 어떤 필드에 어떤 입력을 하여야 하며 어떤 버튼을 또는 어떤 링크를 눌러야 하는지 알아내는 것이다. 웹 테스트를 자동화하려면 WinRunner와 같은 도구가 필요하며 테스트를 위하여 스크립트로 표현하면 다음과 같다.

```
web_browser_invoke(IE,
  "c:/sss_test/vantagepoint.html";
set_window("VantagePointNetwork: Home", 5);
web_link_click) Get Started Today );
set_window("VantagePoint Network: Members Sign In", 4):
edit_set("firstName", "Kevin");
edit_set("lasttName", "Muir")
edit_set("businessName", "My Firm")
edit_set("add1", "1 Any Street")
edit_set("city", "Fort Collins")
list_select_item("region", "Colorado");
edit_set("zip", "80525")
edit_set("phone", "970 221-1234");
list_select_item("paymentType", "Annual Invoice");
button_set("FreePromo", ON);
edit_set("username", "muir99");
edit_set("pass1", "kevin2");
edit_set("pass2", "kevin2");
edit_set("keyword", "fname+2");
button_press("Next->");
```

스크립트로 표시한 특정한 액션이 이루진 후 사용자에게 주어지는 출력은 웹페이지나 다이얼로그박스로 제공된다. 웹기반 소프트웨어의 규모가 크다면 일일이 입력을 주고 상태 변환과 출력 결과를 점검하는 것은 매우 많은 노력과 시간을 소요한다. 따라서 테스트 자동화 도구가 필요하다.

6. 웹 기반 소프트웨어 시험 도구

웹기반 소프트웨어는 매우 다양한 환경에서 수행된다. 다양한 테스트 환경, 즉 다수의 트랜잭션을 발생하거나 GUI를 캡춰하여 플에이백하여 주며 스크립트에 의하여 자동적으로 구동하고 출력을 체크하는 웹 테스트 도구가 사용되고 있다. 웹 테스트 도구의 종류는 앞서 설명한 시험 방법에 따라 다음과 같이 나눌 수 있다. 표 7은 대표적인 도구들을 소개한 것이다.

- 정적분석기 - 네비게이션 링크, 철자법, HTML 호환 오류 등을 찾아내기 위한 도구
- 성능 테스트 도구 - 웹 부하 분석 및 시뮬레이션 도구
- GUI 캡춰 - 웹 사용자 인터액션 과정을 캡춰하고 재생하는 도구
- 런타임 오류 추적 도구
- 웹 보안 시험 도구
- 자바 응용 및 애플릿 시험 도구

표 7 웹 테스트 도구 사례

도구	기능	제조회사	플랫폼	홈페이지
Astra SiteTest LoadRunner SiteManager	스트레스 테스트 웹 통합테스트 비주얼 관리 도구	Mercury Interactive사	Windows	http://www.merc-int.com
SilkTest SilkPerformer SilkStar	기능 및 리그레션 테스트 성능 테스트 Sun 환경에 수행되는 도구	Segue	Windows Unix	http://www.segue.com
SiteRuler Jtest	링크 및 HTML 호환성 체크 Java 프로그램 테스트	ParaSoft	Windows, Linux, Solaris Windows	http://www.parasoft.com
W3C Validator	온라인 HTML 호환성 및 링크 체크	W3C		http://validator.w3.org
VisualTest Performance Studio	GUI 캡춰 성능 테스트	Rational	Windows Windows, Unix	http://www.rational.com
NMap	네트워크 보안(TCP, 패킷, 포트 스캐너)	insecure	Unix	http://www.insecure.org/nmap/

7. 결 론

웹기반 소프트웨어는 구성 요소가 다양하고 자주 변경이 일어나며 다이나믹한 환경에서 수행되는 특수한 소프트웨어라는 점에서 일반적인 소프트웨어의 시험 기술로 커버하지 못하는 오류가 많다. 따라서 웹기반 소프트웨어를 구성하는 다양한 컴포넌트를 분리하여 단위테스트하고 이를 통합하여 시험하되 시험 목적에 따라 다른 기법을 적용하여야 한다.

변경이 자주 일어난다는 측면에서 리그레션 테스트가 중요하며 웹페이지에 포함된 내용의 오류나 링크 오류를 찾기 위한 정적 분석이 필수적이다. 방대하고 분산된 웹기반 소프트웨어를 블랙박스로 테스트하려면 웹페이지를 상태기반으로 나타내고 이벤트에 의한 상태의 변환을 점검하는 방법이 유용하다.

웹기반 소프트웨어의 개발 규모와 빈도가 확대되면서 시험 도구의 사용이 절대적으로 필요하다. 또한 신뢰도 높은 웹사이트를 소수의 인력으로 계속 유지관리하기 위하여 웹시험 기술과 도구에 대한 새로운 연구가 계속 이루어져야 할 것이다.

참고문헌

[1] E. Miller, "WebSite Quality Challenge", White Paper, http://www.soft.com, 2000.

[2] T. Powell, Web Engineering, Prencice-Hall, 1998.

[3] H. Nguyen, Testing Applications on the Web, Wiley, 2001.

[4] M. Cartwright, Empirical Perspectives on Maintaining Web Systems: A Short Review, IEEE Trans. on Software Engineering, vol. 26-8, Aug., pp. 786-796, 2000.

[5] P. Lynch, S. Horton, Web Style Guide: Basic Design Principles for Creating Web Sites, Yale University Press, 1999.

[6] SilkTest: automated functional and regression testing, http://www.segue. com/html/s_solutions/s_silktest/s_silktest_toc.htm

[7] K. C. Bourne, Testing Client/Server System, McGraw-Hill, 1997.

[8] E. Larson, B. Stephens, Web Server, Security, and Maintenance, Prentice-Hall, 2000.

[9] S. Fujiwara, G. Bochmann, et. Al, Test Selection Based on Finite State Machine , IEEE Trans. on Software Engineering, Vol. 17. No. 6, June 1991, pp.591-603.

[10] 강제성, 윤광식, 오승욱, 권용래, "웹의 상태기반 기능시험 기법", 2000 정보과학회 봄 학술발표 논문집, 27권 1호, pp. 501-503, 200.

[11] 권영호, 최은만, "웹 기반 소프트웨어의 테스트 모델에 관한 연구", 2001 정보처리학회 춘계학술발표논문집, 8권 1호, pp.197-200, 2001.

[12] C. Kallepalli, J. Tian, Usage Measurement for Statistical Web Testing and Reliability Analysis, Proceedings of 2001 Software Metrics Symposium, pp. 148 -158, 2001.

최 은 만

1982 동국대학교 전산학과(학사)
1985 한국과학기술원 전산학과(석사)
1993 일리노이 공대 전산학과(박사)
1985~1988 한국표준연구원 연구원
1988~1989 (주)데이콤 주인연구원
1993~현재 동국대학교 컴퓨터공학과 부교수
2000~현재 콜로라도 주립대 교환교수
관심분야 : 객체지향 소프트웨어공학, 소프트웨어 유지보수, 재사용, 역공학

E-mail:emchoi@dgu.ac.kr